

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**

(21) Application No 9804831.7

(22) Date of Filing 06.03.1998

(71) Applicant(s)
Inventec Corporation
(Incorporated in Taiwan)
Inventec Building, No 66 Hou-Kang Street,
Shih-Lin District, Taipei City, Taiwan

(72) Inventor(s)
Xiao-Gang Hu
Jackson Chang

(74) Agent and/or Address for Service
Haseltine Lake & Co
Imperial House, 15-19 Kingsway, LONDON,
WC2B 6UD, United Kingdom

(51) INT CL⁶
G06F 17/30 3/02

(52) UK CL (Edition Q)
G4A AKS AUB

(56) Documents Cited
US 5337347 A US 4677659 A

(58) Field of Search
UK CL (Edition P) G4A AKS AUB
INT CL⁶ G06F 3/02 3/023 17/30

(54) Abstract Title
Searching a database through a simplified keyboard

(57) Use of a simplified keyboard such as a CCITT keyboard (shown) to search a database comprising a plurality of records each including at least one index item for Inquiry (e.g. "NAME", fig. 3). The user inputs an input string by pressing the keys (S1, fig. 4). Each input key of the input string is compared with the same-positioned characters of the index items (S2, fig. 4) for locating data records whose index items match the input string. During the comparison step, the data records identified by matching the first inputted key are stored in a searching buffer, S25, and subsequent comparisons are carried out based on the content of this buffer, S27. The matched data records, from which the user can select, are displayed on a display device (S3, S4, fig. 4). Search speed can be improved by utilising a "key area" part of the character storage codes (10a, fig. 6). The need for using repeated key presses to input single characters (fig. 2) is obviated.

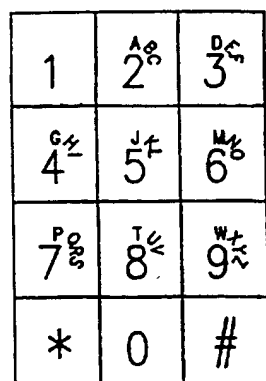


FIG. 1

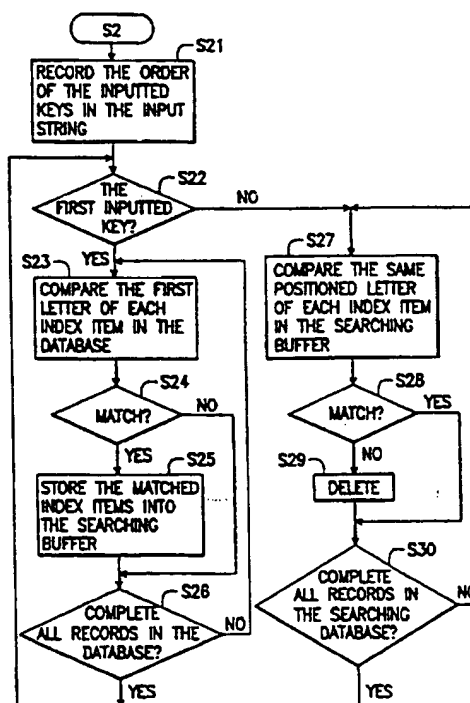


FIG. 5

1	$2^{\text{A} \circ}$	$3^{\text{D} \text{E}}$
$4^{\text{G} \text{H}}$	$5^{\text{J} \text{K}}$	$6^{\text{M} \text{N}}$
$7^{\text{P} \text{Q} \text{R} \text{S}}$	$8^{\text{T} \text{U} \text{V}}$	$9^{\text{W} \text{X} \text{Y} \text{Z}}$
*	0	#

FIG. 1

KEY-IN LETTER	PRESSING KEYS	KEY_PRESSING NUMBER
"A"	"2"+ RIGHT_SHIFT	2
"N"	"6"+"6"+ RIGHT_SHIFT	3
"D"	"3"+ RIGHT_SHIFT	2
"E"	"3"+"3"+ RIGHT_SHIFT	3
"R"	"7"+"7"+"7"+ RIGHT_SHIFT	4
"S"	"7"+"7"+"7"+"7"+ RIGHT_SHIFT	5
"O"	"6"+"6"+"6"+ RIGHT_SHIFT	4
"N"	"6"+"6"	2
TOTAL		25

FIG. 2

NAME(index)	TELEPHONE
...	...
ALEX	2567-3456
ANDERSON	2345-7654
CATHY	2703-9911
JONATHAN	2225-9445
PETER	2704-1457
...	...

FIG. 3

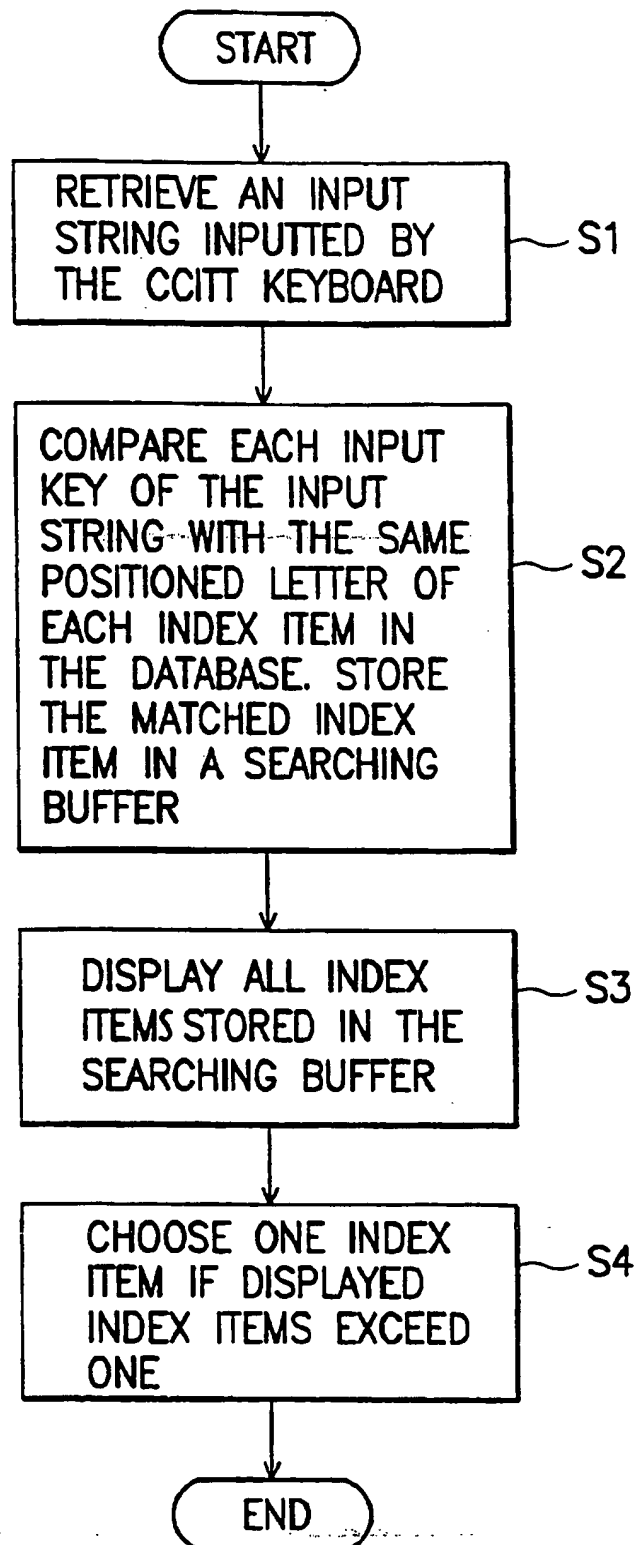


FIG. 4

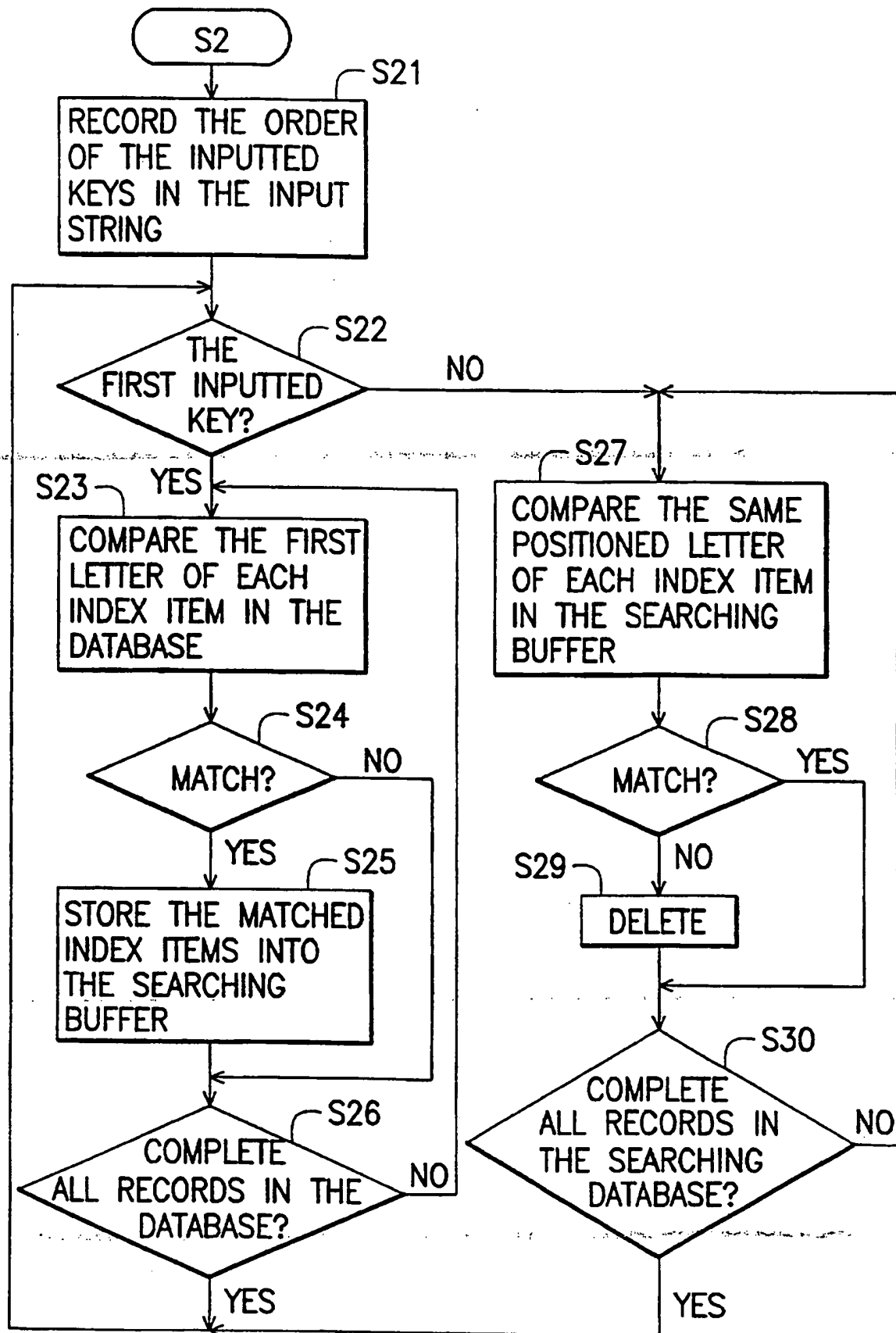


FIG. 5

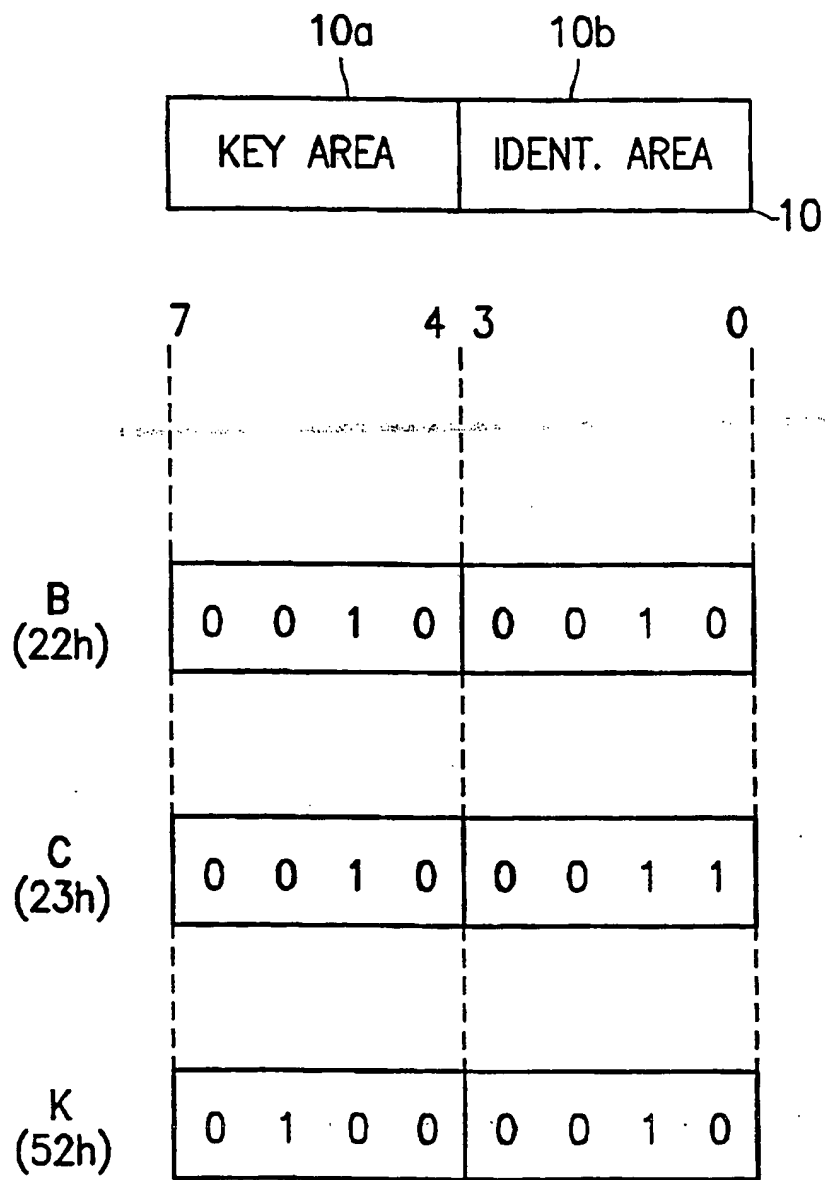


FIG. 6

"NAME" INDEX ITEM	STORAGE CODES (HEX)
...	...
ALEX	21 53 32 92
ANDERSON	21 62 31 32 73 74 63 62
CATHY	23 21 81 42 93
JONATHAN	51 63 62 21 81 42 21 62
PETER	71 32 81 32 73
...	...

FIG. 8

SEARCHING A DATABASE THROUGH A SIMPLIFIED
KEYBOARD

The present invention relates in general to a
5 searching method for a database, more specifically, to a
method for querying or searching a database through a
simplified keyboard, such as the keyboard arrangement
defined by the Consultative Committee for International
Telephony and Telegraphy (CCITT).

10 CCITT keyboards are widely employed in telephone sets
and some handset electronic notebooks as input devices.
Compared with the normal keyboard used in computers, the
CCITT keyboard can be regarded as a simplified one. FIG. 1
(Prior Art) shows the key arrangement defined in the CCITT
15 keyboard. As shown in FIG. 1, there are twelve keys in the
CCITT keyboard, including numeral keys "0"-"9" and special
keys "*" and "#". In addition, there are three or four
alphabetical letters marked upon the upper portion for each
of the numeral keys "2"-"9". For example, the numeral key
20 "2" corresponds to the letters "A", "B" and "C"; the
numeral key "3" corresponds to the letters "D", "E" and
"F", and so on. Therefore, in the CCITT keyboard, any one
of the numeral keys "2"-"9" can be depressed to input
multiple alphabetical letters.

25 Generally speaking, there are two approaches to
inputting the alphabetical letters marked upon the
corresponding keys through the CCITT keyboard. The first
approach is to directly input according to the relationship
between the numeral keys and the corresponding alphabetical
30 letters. Such a manner is used for users to memorize a
telephone number by means of a meaningful word. When
dialing the telephone number, the users can directly press
the numeral keys corresponding to each letter of the word.
Note that it is not required for the telephone or switching

equipment to provide any additional processing for the dialing.

Another approach is used for inputting real data. Conventionally, inputting the real alphabetical letter through the CCITT keyboard is complicated and mainly achieved by a manner of repeatedly depressing the corresponding numeral key. For example, in the key arrangement of the CCITT keyboard, the letters "A", "B" and "C" are marked upon the upper portion of the numeral key "2". Therefore, if the user desires to input a letter "A", he or she should press the numeral key "2" once. In the similar manner, pressing twice the key "2" represents inputting a letter "B" and pressing three times the key "2" represents inputting a letter "C". In addition, if the user desires to input the next letter, he or she should press a LEFT-SHIFT key once and input the next letter in the same manner. Usually, the "*" key is defined as the RIGHT-SHIFT key and the "#" key is defined as the LEFT-SHIFT key.

FIG. 2 (Prior Art) illustrates an example of keying the name "ANDERSON" using the conventional input scheme. In FIG. 2, the input letter, the pressed keys on the CCITT keyboard and the key-pressing number are respectively shown for each letter of the name "ANDERSON." Referring to FIG. 2, when inputting the name, "ANDERSON," containing eight letters using a CCITT keyboard, the user must at least press twenty-five keys. Therefore, it is inconvenient for the operator to input real data. Since this process is both time-consuming and easily subject to error, therefore, providing a means to effectively input data through a simplified keyboard, such as the CCITT keyboard or the like, is a major objective of the present invention.

According to the above description, an object of the present invention is to provide a method for querying or searching a database using the CCITT keyboard or the like to facilitate the operation. In other words, this novel method can lessen the time required to depress keys during the inquiry and searching process and thus can reduce the overall operation time.

The present invention achieves the above-indicated objects by providing a method for searching a database through the use of a simplified keyboard, such as the CCITT keyboard. The database has a plurality of data records and each data record has an index item stored by letters. In addition, the number of the letters is greater than the number of the keys on the simplified keyboard and each key is used to input one or more than one letter. First, the user presses the keys on the simplified keyboard for inquiry and the input keys constitute an input string. Next, each input key of the input string is compared with the same positioned letters of the index items of the data records for locating the data records having the index items that match the input string. During the comparison step, the located data records are stored in a searching buffer and displayed on a display device. If the number of the displayed index items exceed one, the user can readily choose one to complete the searching operation.

In addition, the storage code of each letter in the database includes a key area that one-to-one corresponds to the key for inputting the letter. Therefore, in the step of comparing the input string and the index items, the key area can be used for determining whether the index item matches with the input string. This can improve the searching speed.

In the accompanying drawings:

FIG. 1 (Prior Art) shows the key arrangement defined in the CCITT keyboard.

5

FIG. 2 (Prior Art) illustrates an example of keying the name "ANDERSON" using the conventional input scheme.

FIG. 3 illustrates an example of a telephone book database in the embodiments of the invention.

10

FIG. 4 shows a flowchart of the inquiring and searching procedure in the first embodiment of the present invention.

15

FIG. 5 shows a flowchart of the detailed steps in the comparison step of FIG. 4.

FIG. 6 illustrates the storage code format of the letters according to the second embodiment of the present invention.

20

FIG. 7 shows a storage code table that defines the letters and their corresponding storage codes in the second embodiment, where the letters include alphabetical letters, number letters and special letters.

25

FIG. 8 shows the index items in the database shown in FIG. 3 and their storage codes according to the storage code table in FIG. 7.

30

The present invention deals with the case of querying a database through a CCITT keyboard or other simplified keyboards. As described above, the characteristic of the CCITT keyboard or the like is to use fewer keys on the

35

keyboard (or the corresponding scanning codes generated by pressing the keys) to input a large number of symbols, such as numbers and alphabetical letters. For example, the CCITT keyboard employs eight numeral keys "2"-"9" to input
5 twenty-six alphabetical letters. Therefore, each of these numeral keys should correspond to one or more than one alphabetical letters. The present invention adopts a fuzzy input approach to handle the key-in process during the database inquiry. That is, the data input by the user for
10 querying the database is fuzzy or uncertain. However, the present invention provides a handling procedure to deal with the fuzzy or uncertain inquiry data and to search for a match entry in the database. The following embodiments employ the CCITT keyboard for illustrating the invention.
15 However, it is understood by those skilled in the art that the processing method in the following embodiments also can be applied to other cases of using a simplified keyboard as an input device for inquiry.

20 First Embodiment:

At the beginning, the database ready for inquiry by a CCITT keyboard is described. As described above, the CCITT keyboard or other similar keyboards serve as input devices
25 in the telephone sets and the handset electronic notebooks for querying the database therein. Usually, the database system in these devices is not a massive one and has a simple database structure, such as a telephone book or address book. Generally speaking, the database comprises
30 many data records. Each of the data records includes at least one index item for inquiry. For example, each record of a telephone book includes two items. FIG. 3 illustrates an example of a telephone book database used in the
embodiments of the invention. As shown in FIG. 3, each
35 data record contains an index item "NAME" 2 and an item

"TELEPHONE" 4. While searching such a database, users should input a query for searching the desired record in the database.

5 This embodiment and the conventional case are different in the manner of inputting queries and the searching method during the whole searching process inquired by the CCITT keyboard. The difference in the manner of inputting queries is first described. In the
10 conventional case, the user must repeatedly press a key once or more times to input one corresponding alphabetical letter. Pressing two keys for inputting one alphabetical letter is the minimum requirement, that is, pressing the corresponding numeral key and pressing the RIGHT-SHIFT key.
15 On the other hand, in this embodiment, the users only press a numeral key for inputting an alphabetical letter. Consider the case of inputting the name "ANDERSON." Using the conventional input manner, it is necessary for the user to press at least twenty-five keys in order to input the
20 word "ANDERSON." However, the user that employs the input manner in this embodiment need only press eight corresponding numeral keys, that is, "26337766."

The second difference between the embodiment and the
25 prior art is the searching process. FIG. 4 shows a flowchart of the querying and searching processes in this embodiment of the present invention. At first, the system retrieves an input string input by the user through the CCITT keyboard (S1). The input string contains at least
30 one number directly input by pressing the corresponding numeral key on the CCITT keyboard. For example, if the user attempts to input a name "ANDERSON" for querying the telephone book database, as shown in FIG. 3, he or she should sequentially press the following numeral keys:

26337766. Then the numeral string "26337766" serves as the input string in the following steps.

According to the input string, the system can compare
5 each input key of the input string with the same positioned
letter of each index item in the database (S2). The query
name "ANDERSON" and the input string "26337766" are applied
to the database shown in FIG. 3 for explanation. The first
input key "2" of the input string is used to compare with
10 the first letter of each index item in the database.
According to the key arrangement defined by the CCITT
keyboard, the key "2" corresponds to the alphabetical
letters "A", "B" and "C." Therefore, the system locates
the index items containing a first alphabetical letter of
15 "A", "B" or "C." In the database shown in FIG. 3, the
index items "ALEX", "ANDERSON" and "CATHY" can meet the
requirement, but the index items "JONATHAN" and "PETER"
cannot. The second input key "6" of the input string is
used to compare the second alphabetical letter of each
20 index item satisfying the requirement of the first input
key. In this case, the remaining index items are "ALEX,"
"ANDERSON" AND "CATHY." The key "6" in the CCITT keyboard
corresponds to the letters "M", "N" and "O." Thus, only
the index item "ANDERSON" can meet the requirement with
25 respect to the second input key. During the comparison
process, the matched index items are stored in a searching
buffer (S2). The searching buffer is used for storing the
related information about the matched records in the
database. Therefore, only the index items of the data
30 records satisfying the requirement of the input string can
be stored in the searching buffer. In addition, in this
embodiment, if none of the index items can meet the
requirement of the first input key, the system will
download all index items of the data records in the
35 database to the searching buffer and display them in a

display device for further processing, such as selection, browse or quit.

5 Next, the system displays all of the index items stored in the searching buffer on a display device (S3). Therefore, when the user presses the keys for querying, the temporary searching result will be immediately shown on the display device for reference. Because the input query is uncertain or fuzzy in this embodiment, the database system
10 may locate one or more than one data records corresponding to the currently inputted string. Therefore, it is not definitely sure whether the preferred data record queried by the user can be directly found during such a searching process. However, in such a simple database or even in a
15 more complex database, the possibility of a large amount of index items matching with the input string is still low. If the number of the displayed index items exceeds one, the user can readily choose one to complete the searching operation (S4).

20

 The key step shown in the process flowchart of FIG. 4 is the comparison step S2, which will be further described hereinafter. FIG. 5 shows a flowchart of the detailed steps in step S2 of FIG. 4. In step S1 shown in FIG. 4,
25 the user may press the keys on the CCITT keyboard to input the input string. Therefore, at the beginning of the comparison, the system may record the keyed-in order of each input key in the input string (S21). In the following step, the first input key and other input keys of the input
30 string will be separately processed. Before the first input key is processed, the searching buffer is empty. Therefore, the comparison of the first input key is performed in the database. Meanwhile, the contents of the searching buffer is established during the comparison
35 process with respect to the first input key. After the

contents of the searching buffer are established, the following comparison processes with respect to other input keys are performed in the searching buffer.

5 Returning to FIG. 5, if the currently processed key of the input string is the first input key (S22), then the steps S23-S26 are performed. At first, the first input key is sequentially compared with the first letter of each index item in the database (S23). If the first input key
10 of the input string and the first alphabetical letter of an index item are matched (S24), the matched index item is stored in the searching buffer (S25). These comparison and storage steps are continuously performed until all records in the database are completed (S26). Finally, all index
15 items having a first alphabetical letter matched with the first input key of the input string are stored in the searching buffer. Then the comparison process for other input keys of the input string can be directly performed in the searching buffer.

20

 If the currently processed key of the input string is not the first input key (S22), then the steps S27-S30 are performed. The procedure of processing the other input keys is similar to that of processing the first input key.
25 The difference between them is the searching target. The searching process with respect to the first input key (S23-S26) is performed on the whole database, but the searching processes with respect to the other input keys are performed on the searching buffer. The currently
30 processed input key is sequentially compared with the same positioned alphabetical letter of each index item stored in the searching buffer (S27). If the currently processed input key and the same positioned alphabetical letter of an index item are not matched (S28), this index item is
35 deleted from the searching buffer (S29). Therefore, only

the index items satisfying the requirement can remain in the searching buffer. These comparison and storage steps are continuously performed until processing of all records in the searching buffer is completed (S30). In steps
5 S27-S30, the index items that cannot conform to the pattern of the input string may be deleted from the searching buffer.

As described above, the contents of the searching
10 buffer can be refined in response to each input key of the input string. In addition, the contents of the searching buffer can be displayed on the display device in step S3 of FIG. 4. Therefore, the user can preview and realize the temporary searching result during inputting the inquiry
15 data. In addition, it is possible in advance to find out the desired data entry before the input is completed.

In this embodiment, the user can easily input the query for searching and supervise the whole searching
20 procedure. Therefore, the querying method in this embodiment can enhance the capability of the CCITT keyboard in searching. Consider the case of a name "ANDERSON" is input for inquiry through the CCITT keyboard. Table 1 shows the comparison result between the prior art and the
25 first embodiment in this case.

Table 1

COMPARISON ITEM	THE PRIOR ART	THE FIRST EMBODIMENT
NUMBER OF KEYS PRESSED	25	8
INPUT PROPERTY	DEFINITE	FUZZY
OPERATION	COMPLEX, MISTAKE- PRONE	EASY

As shown in Table 1, the first embodiment of the present invention is superior to the prior art in reducing the number of keys that must be depressed for inputting the querying data. In addition, the scheme for inputting a query becomes straightforward, thus reduces the possibility that the user will press the wrong keys. Although the input data in this embodiment is fuzzy, the system still can ferret out the desired data entry in the address book or the telephone book.

Second Embodiment:

The present embodiment is the same as the first embodiment except for the storage format of the alphabetical letters in the database. In many personal computers, the most popular system for coding alphabetical letters is the American Standard Code for Information Interchange (ASCII). Original ASCII uses seven bits to represent one letter, but now eight-bit extensions are widely applied to various operation systems. In the ASCII coding scheme, the alphabetical letters are sequentially encoded. For example, in the ASCII coding, codes from 41(hex) through 5A(hex) represent the capital letters "A" through "Z," respectively. In addition, codes from 61(hex) through 7A(hex) represent the lower cases "a" through "z."

However, using the ASCII format to represent the contents of the database may complicate the comparison process between the input string and the index items in the first embodiment. Consider the case of the input string containing a first input key "2." According to the key arrangement of the CCITT keyboard, the numeral key "2" corresponds to the letters "A," "B" and "C." Therefore, the first letter of each index item in the database should

be individually compared with the ASCII codes for letters "A," "B" and "C," which determines whether the currently processed index item can meet the requirement with respect to the currently processed input key. In more detail, the
5 ASCII code for the first letter of the index item ready for comparing should be individually compared with 41(hex), 42(hex) and 43(hex), which represent the ASCII codes for the letters "A," "B" and "C." In other words, three or four comparison steps are required for processing one index
10 item. Therefore, such a comparison manner may reduce the speed of the data search. In this embodiment, the storage coding of the alphabetical letters in the database is modified to improve the searching speed.

15 Referring to FIG. 1, the CCITT keyboard has twelve keys defined in the keypad, including ten numeral keys (0 through 9) and two special symbol keys ("*" and "#"). In this embodiment, the storage codes of all alphabetical letters "A" through "Z" are re-defined with respect to the
20 ten numeral keys "0" through "9" in the CCITT keyboard.

FIG. 6 illustrates the storage code format of the letters according to the second embodiment of the present invention. The common storage code format still has eight
25 bits, which are respectively denoted by bit 0 through bit 7 from right to left, as shown in FIG. 6. In addition, each storage code 10 can be divided into two parts, which are key area 10a and serial area 10b. In this embodiment, key area 10a comprises four bits, from bit 7 to bit 4. Serial
30 area 10b also comprises four bits, from bit 3 to bit 0. Some storage code examples are also illustrated in FIG. 6. For example, the storage code of the alphabetical letter "B" is 22(HEX). The storage code of the letter "B" can be expressed as 00100010 in the binary form, where the former
35 four bits serve as the key area 10a and the latter four

bits serve as the serial area 10b. In addition, the storage code of the alphabetical letter "C" is 23(HEX), or 00100011 in the binary form. Apparently, the storage codes of the alphabetical letters "B" and "C" have the same code in the key area 10a, but are different in the serial area 10b. In addition, the storage code of the alphabetical letter "K" is 52(HEX), or 01000010 in the binary form. Apparently, the code in key area 10a of the alphabetical letter "K" is different from that of the alphabetical letters "B" and "C."

Key area 10a is used to identify the correspondence between the letter and the key in the CCITT keyboard. All letters corresponding to the same key defined in the CCITT keyboard have the same code in the key area 10a. The code in the serial area 10b is used to distinguish the letters corresponding to the same key defined in the CCITT keyboard. For example, all alphabetical letters corresponding to the key "2," such as "A," "B" and "C," should have the same code in the key area 10a. In this embodiment, this code in the key area 10a is defined as 2(HEX). In addition, the codes in the serial area 10b of the alphabetical letters "A," "B" and "C" are different, that is, 1(HEX), 2(HEX) and 3(HEX), respectively. FIG. 7 shows the storage codes of the letters defined in this embodiment, including the alphabetical letters, number letters and special letters. It is noted that the key codes of the letters corresponding to the keys "2" through "9" in the CCITT keyboard are defined as 2(HEX) through 9(HEX), respectively.

When the data in the database, especially the index items of the data records, are stored by the storage code format shown in FIG. 7, the time for comparing each index item with the input string can be dramatically reduced.

For example, if the input key is "2," the judgement can be made by determining whether the key area of the storage code of the same positioned letter is 2(HEX) or not. It means that only one comparison step with respect to an index item is required. Therefore, using such a storage code format in the database can reduce the comparison time and increase the searching speed.

Referring to the database shown in FIG. 3 and the inquiry procedure shown in FIG. 5, an example of querying and searching the database employing the storage code format in this embodiment is illustrated. In this example, the searching name is "ANDERSON," and, therefore, the user will sequentially press the following numeral keys, "26337766." FIG. 8 shows the "NAME" index items in the database and their storage codes according to the storage code table in FIG. 7.

When the user presses the first input key "2," the steps S23~S26 in FIG. 5 that compare the first letters of all index items with the input key "2" are performed. These comparison steps can be easily performed. That is, any index item that has the storage code of the first letter to be 2X (X can be any hexadecimal digit) can meet the requirement with respect to the first input key. In FIG. 8, the index items "ALEX," "ANDERSON" and "CATHY" can meet the requirement and will be downloaded to the searching buffer. When the user presses the second input key "6," the steps S27~S30 in FIG. 5 are performed. Any index item in the searching buffer that has the storage code of the second letter to be 6X (X can be any hexadecimal digit) can meet the requirement with respect to the second input key. In this example, only the index item "ANDERSON" can meet all requirements, and other index items will be deleted from the searching buffer. At this time,

the user has located the desired data record. According to the above description, only one comparison step for determining whether any index item corresponds to the current input key is required. Compared with the first embodiment, the second embodiment can reduce the number of comparison operations.

In this embodiment, the storage code format shown in FIG. 6 is designed for the CCITT keyboard. That is, the key area 10a should represent different numeral keys "0" through "9" in the CCITT keyboard. Therefore, it is required that the bit size of the key area 10a should be at least 4 since $2^3 = 8 < 10$. In this embodiment, the bit size of the key area 10a is set to be 4, but also can be set to 5 or 6. It is understood by those skilled in the art that the bit sizes of the key area 10a and the serial area 10b can be adjusted according to the practical situation in other cases of using the simplified keyboard.

Consider a case that an inquiry name "ANDERSON" is input for inquiry through the CCITT keyboard. Table 2 in the following shows the comparison result between the prior art and the second embodiment in various aspects.

Table 2

COMPARISON ITEM	THE PRIOR ART	THE SECOND EMBODIMENT
NUMBER of KEYS DEPRESSED	25	8
INPUT PROPERTY	DEFINITE	FUZZY
OPERATION	COMPLEX, MISTAKE- PRONE	EASY
COMPARISON TIME	ONCE FOR EACH INDEX ITEM	ONCE FOR EACH INDEX ITEM

Just as with the first embodiment, the second embodiment is superior to the prior art in terms of the number of keys that must be depressed and the facility of operation. In addition, the comparison times of the searching process in the second embodiment are reduced. Therefore, the present invention is superior to the prior art in the case of using a CCITT keyboard for querying the database.

10

While the invention has been described by way of example and in terms of the preferred embodiment, it is to be understood that the invention is not limited to the disclosed embodiments. On the contrary, it is intended to cover various modifications and similar arrangements as would be apparent to those skilled in the art. Therefore, the scope of the appended claims should be accorded the broadest interpretation so as to encompass all such modifications and similar arrangements.

15

WHAT IS CLAIMED IS:

1 1. A method for searching a database through a
2 simplified keyboard, the database having a plurality of
3 data records and each data record having an index item
4 stored by letters, the simplified keyboard having a
5 plurality of keys for inputting a query for the database,
6 wherein the number of the letters is greater than the
7 number of the keys and each key corresponds to at least one
8 letter, the method comprising the steps of:

9 receiving an input string input by pressing the keys
10 on the simplified keyboard, the input string being composed
11 of at least one element corresponding to one input key;

12 sequentially comparing each input key element of the input
13 string with the same positioned letters of the index items
14 of the data records, for locating the data records having
15 the index items that match with the input string;

16 displaying related information of the located data
17 records on a display device; and

18 selecting one of the located data records.

1 2. The method of claim 1, wherein the storage code of
2 each letter includes a key area that corresponds one-to-one
3 to the key for inputting the letter.

1 3. The method of claim 2, wherein the step of
2 comparing the input string and the index items uses the key
3 area for determining whether the index item matches with
4 the input string.

1 4. The method of claim 3, wherein the simplified
2 keyboard is the CCITT keyboard.

1 5. The method of claim 1, where the step of comparing
2 the input string and the index items comprises the steps
3 of:
4 defining a searching buffer;
5 comparing a first input key of the input string with
6 the first letters of the index items of the data records,
7 for locating the data records having the index items that
8 match with the first input key of the input string;
9 storing the index items of the located data records in
10 the searching buffer;
11 sequentially comparing other input keys of the input
12 string with the same positioned letters of the index items
13 stored in the searching buffer when the input string has
14 more than one input key; and
15 deleting the index items that do not match with the
16 input string from the searching buffer.

1 6. The method of claim 5, further comprising a step
2 of downloading all of the index items in the searching
3 buffer when none of the data records has the index item
4 that matches with the first input key of the input string;

1 7. The method of claim 5, wherein the storage code of
2 each letter includes a key area that corresponds one-to-one
3 to the key corresponding to the letter, and in the step of
4 comparing the input string and the index items, the key
5 area is used for determining whether the index item matches
6 the input string.

1 8. The method of claim 5, wherein the index items
2 stored in the searching buffer are displayed in the
3 displaying step.

1 9. The method of searching a database through a
2 simplified keyboard substantially as hereinbefore described

with reference to and as shown in the accompanying drawings.

- 5 10. A database system comprising:
- a database having a plurality of data records each
 having an index item stored as characters,
 a simplified keyboard having a plurality of keys,
10 less than the number of possible characters, each
 character being associated with one key, for inputting
 a query for the database,
 a display device, and
 a comparison device adapted to compare an input
15 query, generated by a sequence of keystrokes, with the
 corresponding sequence of characters of the index items
 of the data records, the display device being adapted
 to display any matches thus found by the comparison
 device.



Application No: GB 9804831.7
Claims searched: 1-10

Examiner: Melanie Gee
Date of search: 26 August 1998

Patents Act 1977
Search Report under Section 17

Databases searched:

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:

UK CI (Ed.P): G4A (AKS, AUDB)

Int CI (Ed.6): G06F 3/02, 3/023, 17/30

Other:

Documents considered to be relevant:

Category	Identity of document and relevant passage	Relevant to claims
X	US 5337347 A (HALSTEAD-NUSSLOCH et al.), see whole document.	1 & 10
X	US 4677659 A (DARGAN), see col. 4 line 20 - col. 7 line 39.	1 & 10

X Document indicating lack of novelty or inventive step
Y Document indicating lack of inventive step if combined with one or more other documents of same category.
& Member of the same patent family

A Document indicating technological background and/or state of the art.
P Document published on or after the declared priority date but before the filing date of this invention.
E Patent document published on or after, but with priority date earlier than, the filing date of this application.